

**PCT**WORLD INTELLECTUAL PROPERTY ORGANIZATION  
International Bureau

## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<b>(51) International Patent Classification <sup>5</sup> :</b> <b>G06F 9/44</b>	<b>A1</b>	<b>(11) International Publication Number:</b> <b>WO 91/17497</b> <b>(43) International Publication Date:</b> 14 November 1991 (14.11.91)
<b>(21) International Application Number:</b> PCT/GB91/00687 <b>(22) International Filing Date:</b> 30 April 1991 (30.04.91) <b>(30) Priority data:</b> 9009702.3 30 April 1990 (30.04.90) GB <b>(71) Applicant (for all designated States except US):</b> HEWLETT-PACKARD COMPANY [US/US]; 3000 Hanover Street, Palo Alto, CA 94304 (US). <b>(72) Inventor; and</b> <b>(75) Inventor/Applicant (for US only) :</b> ROBSON, Christopher [GB/GB]; Hewlett-Packard Limited, Filton Road, Stoke Gifford, Bristol BS12 6QZ (GB). <b>(74) Agent:</b> SMITH, Martin, Stanley; Stevens Hewlett & Perkins, 1 St. Augustine's Place, Bristol BS1 4UD (GB).		<b>(81) Designated States:</b> AT (European patent), BE (European patent), CH (European patent), DE (European patent), DK (European patent), ES (European patent), FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, LU (European patent), NL (European patent), SE (European patent), US.  <b>Published</b> <i>With international search report.</i> <i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>
<b>(54) Title:</b> OBJECT BASED COMPUTER SYSTEM  <b>(57) Abstract</b>  An object based computer system has the facility to allow multiple viewers to view and manipulate a semantic object concurrently. The semantic object has a table of viewers which are viewing it and all such viewers are sent update messages to allow them to indicate the set of viewers which is concurrently viewing the object.		

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	ES	Spain	MG	Madagascar
AU	Australia	FI	Finland	ML	Mali
BB	Barbados	FR	France	MN	Mongolia
BE	Belgium	GA	Gabon	MR	Mauritania
BF	Burkina Faso	GB	United Kingdom	MW	Malawi
BG	Bulgaria	GN	Guinea	NL	Netherlands
BJ	Benin	GR	Greece	NO	Norway
BR	Brazil	HU	Hungary	PL	Poland
CA	Canada	IT	Italy	RO	Romania
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland	KR	Republic of Korea	SN	Senegal
CI	Côte d'Ivoire	LI	Liechtenstein	SU	Soviet Union
CM	Cameroon	LK	Sri Lanka	TD	Chad
CS	Czechoslovakia	LU	Luxembourg	TG	Togo
DE	Germany	MC	Monaco	US	United States of America
DK	Denmark				

- 1 -

OBJECT BASED COMPUTER SYSTEM

The invention relates to an object based computer system.

5       Object based computer systems facilitate multi-tasking and multi-user operations where common data files may be accessed concurrently by different tasks operated from the same terminal or operated by different users employing respective terminals. A  
10 windows environment is generally employed in conjunction with an object based system to enable windows to be allocated to respective concurrent tasks. There follows an overview of an object based system in a windows environment which will help define  
15 terms used herein.

An OBJECT is a combination of data and method code which is normally stored on disk. An object may be INACTIVE, when it is identified simply as a disk file, or ACTIVE, when it has a PROCESS, or executable  
20 file, associated with it. When active, at least parts of the object are held in the computer RAM and the object is defined by the state of the associated process rather than by its file.

Objects can be LINKED to other objects so that  
25 changes in one are reflected in the object or objects with which it is linked by virtue of MESSAGES which are passed between the linked objects. The system is controlled by an OBJECT MANAGER which is an application running in the windows environment and  
30 which controls activation and deactivation of objects, and the passage of messages between objects.

An object is sometimes a CONTAINER which contains as notional parts other objects. Examples of container objects in a distributed office system are a  
35 desktop, folder and a document. A VARIABLE DIMENSION

- 2 -

DATA OBJECT (VDO) is a data store of which the values of elements may be viewed and changed directly by a user.

5       Object based systems have suffered the disadvantage common to multi-tasking or multi-user systems sharing data files. This is that although, with careful record-locking procedures, different tasks or users can, in concurrent sessions, alter data in the same data file (or object) each alteration and  
10       display is effectively carried out independently. The users do not have immediate cross-references and updating to changes made by other users to the data file at the same time.

      We have devised a solution to this problem which  
15       depends on splitting an object between a SEMANTIC part (which defines the state of the object) and a PRESENTATION part (for presenting to a user the state of the object). Indeed, conceptually one can think of there being separate semantic objects and presentation  
20       objects. In addition, when using a windows user interface, there are windows for viewing objects and facilitating multi-tasking. In this specification, the presentation part or presentation object is something which is utilised by a window and forms part  
25       of the window for the time in which the window is viewing the object in question and generally a distinction between the window and the presentation part or object will not be made.

      Semantic objects have data stored in a particular  
30       storage domain. A storage domain may be regarded as closely equivalent to a storage medium such as a hard disc or floppy disc in the sense that all objects in a given storage domain are on-line together or are off-line together. Consequently, a single machine may  
35       support a plurality of storage domains.

- 3 -

The present invention could be applied in a single computer having one or more storage domains but is primarily concerned with an object based system having a plurality of user stations. Such a system may be provided by a single central processing device having a plurality of user stations coupled to it, or it may be provided by a distributed processing network consisting of a number of independent processing units each having a respective station associated with it. In the system of the present invention an object has an implicit presentation, which will look the same however viewed.

It is possible to open one or more windows on each object. The windows manage the display and input/output (lexical) interaction in the system. The user of multiple windows in conjunction with an object enables employment of the techniques of sharing (multiple windows to a common object) and distributed (window on one machine and object on another) applications.

We define a VIEWER as an object which allows users to access other objects by providing the appearance of the object to the user and accepting user input for data manipulation. Viewers have hitherto been known as TASK WINDOWS, but a window is part only of the viewer. The object being viewed controls its own state and provides the methods for access to and manipulation of that state.

When sharing or multi-user operations are being effected, each task or user associated with a particular object is a REFEREE of that object. It is ensured that each active object retains references to its referees so that they are sent messages to be notified of changes during object processing.

The semantic/presentation split allows multiple

- 4 -

viewers to be attached to an object. It is important to remember in this case that all viewers are seeing the same object, not a version of it. This is important, as we have a basically 'physical' model of the object world, when a user manipulates an object he is actually physically changing it for everyone. This could lead to conflict when two or more users are trying to manipulate the same thing.

The present invention seeks to provide a solution.

According to the present invention there is provided an object based computer system having at least one computer with a central processor unit, random access memory and program and data storage facilities, the computer being programmed to run in a windows environment and having an object manager whereby objects are activated, the objects being capable of being linked, objects being subject to a semantic/presentation split whereby multiple viewers can be linked to an object to allow concurrent observation and manipulation, and a presence mechanism being provided whereby referees of an object being viewed are notified of the presence of each other by indications in their respective viewers.

Indication of the presence of concurrent viewers of an object greatly facilitates the user's vision of the system and under-pins the user model which the system establishes. Instead of merely observing the effects of his own manipulations of data the user can be aware that other viewers are present and thus capable of separate manipulation of the same data.

A user always knows what he has requested by way of a manipulation. This is the local feedback provided by the viewer. The underlying object updates all viewers of its state, thus the user will see the

- 5 -

end effect. In such conflict conditions it is not possible to guarantee that the request and the end effect match, but the user will not be misled as to what has happened. One of the important benefits of the presence mechanism is to help explain these situations to the user. The semantic/presentation split also allows viewers to be remote (from the object), without the user losing the benefit of immediate feedback of his manipulations.

10       Users themselves can move from one terminal to another. The identity of the users is obtained from Personal Identity Cards which they use to log on to the system.

15       In a preferred arrangement, on each viewer are listed the names of owners of the viewers viewing the same object. The presence of the displaying viewer is always shown as the first on the list. The presence indicators also give the colour coding of the viewer (each viewer has a border which is a different colour to all other viewers on that machine). The object being viewed keeps a list of current viewers and it is the duty of the object to inform each viewer which other viewers are viewing that object.

25       The invention will further be described with reference to the accompanying drawings, of which:-

Figure 1 is a schematic diagram of an object based computer system embodying the invention;

Figure 2 is a diagram illustrating the user's model of a system which incorporates the invention;

30       Figure 3 is a further diagram illustrating part of the user's model of the system;

Figure 4 is a diagram of a user's view of a system having a presence mechanism in accordance with the invention;

35       Figure 5 is a schematic diagram illustrating the

- 6 -

presence mechanism of the invention;

Figure 6 is a schematic diagram illustrating program procedures of an object being reviewed; and

5 Figure 7 is a schematic diagram illustrating program procedures of a viewer.

Referring to Figure 1 there is shown an object based computer system which comprises a network of personal computers PC, each of which has a central processor unit CPU; random access memory RAM; mass storage facilities MS; a visual display unit VDU and a mouse M. In the drawing these units are designated for one computer only. Each computer also has a card reader CR.

15 The mass storage facilities MS constitute domains in the system which are generally associated with the user or users who habitually use that computer. Thus, objects associated with those users are stored there. These objects include, for example, the desk top objects for each habitually local user.

20 Referring to Figure 2 there is shown a model of a distributed office computer system which comprises several computers linked together in a network as in Figure 1. Each computer has the usual input/output facilities of keyboard, mouse and VDU as well as a CPU and hard disk on which are held object files. The core of the system is the object world 1 containing objects to which the users 2 have access.

25 An important aspect of the system is the model of it which the user has. Critical in this is a layered structure which includes an access layer 3 whereby the users gain access to the objects in the object world 1. Figure 3 shows more specifically the access layer including viewers 4 which are window objects and which are able to be navigated to observe and manipulate different objects in the object world. It is

35



- 7 -

important to note the concept of navigation of a viewer between objects in this system. A user does not need to open a new window for each object. The viewers must be regarded as mechanisms in their own right.

Since a feature of the system is that multiple viewers may be linked to the same object and each viewer has the capability not only to observe but also manipulate the object, it is possible for a user to observe unexpected results. For example, icons might move or data might change. To facilitate an understanding of such effects the present invention provides viewers with indications of concurrent viewers. This is shown in Figure 4, where there is shown a presentation which has a first window which has been opened on "Desktop on IO" and a second window which has been opened on "Folder on IO". Both windows have presence bars 5 and 6 respectively. Bar 5 shows only "self" which indicates that no other viewers are open on the object of that window. Bar 6, on the other hand, shows that in addition to "self" there are the viewers of "Joe" and "George". What is immediately clear, therefore, is that users Joe and George will be viewing "Folder on IO". All three users will observe the same view of "Folder on IO" and will be able to manipulate that object. Each presence indicator has a border of a particular colour corresponding to the viewer in question. The presence of the displaying viewer is always shown as the first in the list. The bar may be switched off by the user to conserve screen space.

Referring to Figure 5 there are shown two viewers 7 and 8 open on the same object 9. Presence bars 10 and 11 on the viewers have presence indicators 12 as described above. However, the object 9 being viewed

- 8 -

has responsibility for maintaining a list 14 of all viewers currently linked to it and of passing messages to the viewers to notify them of each other's presence.

5        Each user has the facility to select for his viewer an object to be viewed, or to remove the object viewed. As a selection is made, an appropriate message is sent from the viewer to the semantic part of the object concerned.

10        Figure 6 shows schematically the procedures in the process of the semantic part 9 of the object of Figure 5. Referring to Figure 6 there are three procedures 15, 16 and 17 which have the self-explanatory names AddViewerToTable; PresenceChange and  
15        RemoveViewerFromTable respectively. Messages from viewers are received and analysed by a switch program structure 18 so that if the message request is to view the object, procedure 15 is implemented whereas if the object is to be removed then procedure 17 is  
20        implemented.

Object 9 keeps a table 14 of viewers which are viewing it. In adding a viewer to the table, procedure 15 uses a data structure which comprises the following fields: number of viewers; user name; viewer  
25        identity; viewer colour; message address and object identity. This data is updated in respect of the new viewer, user and object combination, the number of viewers being incremental by one, for example. The structure is added to an array of such structures  
30        which constitutes the viewer table 14.

Procedure 15 then calls procedure 16 which sends an update message at 19 to all viewers in table 14 to carry information of the table update to the viewers.

In removing a viewer from the table 14, procedure  
35        17 receives a message from the appropriate viewer via

- 9 -

the switch structure 18. The message includes viewer identity and user identity information and this is used to search for the appropriate structure in the array of table 14. If found, the structure is cancelled and the number of current viewers is decremented by one. The revised data structure indicating the number of viewers; user name; viewer identity; viewer colour; and object identity is passed to the change procedure 16 which is implemented to pass appropriate messages to the remaining viewers which are viewing the object.

Figure 7 is a schematic diagram showing program code procedures effective at a viewer to modify the presentation in accordance with the update messages received from the semantic object 9 described above. There is a viewer list 22 which is an array of data structures, each structure carrying information about a viewer which is viewing the object. List 22 is similar in many respects to list 14 of the semantic object 9 (Figure 6). List 22 is updated by virtue of the messages passed from the object 9. Such messages are received by a switch program structure 23 and in accordance with the contents of the message are directed to activate an appropriate procedure. Two messages to which the switch structure responds are of particular significance here and are OBJ2WIN\_PRESENCE and OBJ2WIN\_VIEWERREMOVED respectively

Two procedures 24, 25 are names AddPresence and DeletePresence respectively. In response to the OBJ2WIN\_VIEWERREMOVED message, procedure 25 is invoked. This determines from the message the identity of the viewer to be deleted, searches for it in list 22 and if found deletes it from the list. The effect of the message OBJ2WIN\_PRESENCE is firstly to invoke the DeletePresence procedure 25 and then to

- 10 -

invoke the AddPresence procedure 24 which has the effect of adding a new viewer data structure to list 22. The preliminary deletion of the viewer allows changes in the viewer, for example in size and colour, to be catered for.

5 List 22 can be regarded as a list of buttons, since it is used by a procedure 26 to generate on the screen the presence buttons 12 of Figure 5. It is to be noted that the first button on the list, namely the  
10 left-most button displayed, is always that associated with the immediate viewer.

15

20

25

30

35

- 11 -

CLAIMS

1. An object based computer system having at least one computer with a central processing unit, random  
5 access memory and program and data storage facilities, the computer being programmed to run in a windows environment and having an object manager whereby objects are activated, the objects being capable of being linked, objects being subject to a  
10 semantic/presentation split whereby multiple viewers can be linked to an object to allow concurrent observation and manipulation, and a presence mechanism being provided whereby referees of an object being viewed are notified of the presence of each other by  
15 indications in their respective viewers.
2. An object based computer system as claimed in Claim 1 wherein an object being viewed includes a table of viewers which are viewing it, the table being  
20 updated by messages received from viewers and the object sends update messages, corresponding to changes in the table, to the viewers which are viewing the object.
- 25 3. An object based computer system as claimed in Claim 2 wherein each viewer has a viewer list which is updated by the said update messages to represent the viewers which share the object being viewed.
- 30 4. An object based computer system as claimed in Claim 3 wherein the members of the viewer list generate respective buttons which indicate their identities and which are displayed on the viewer.

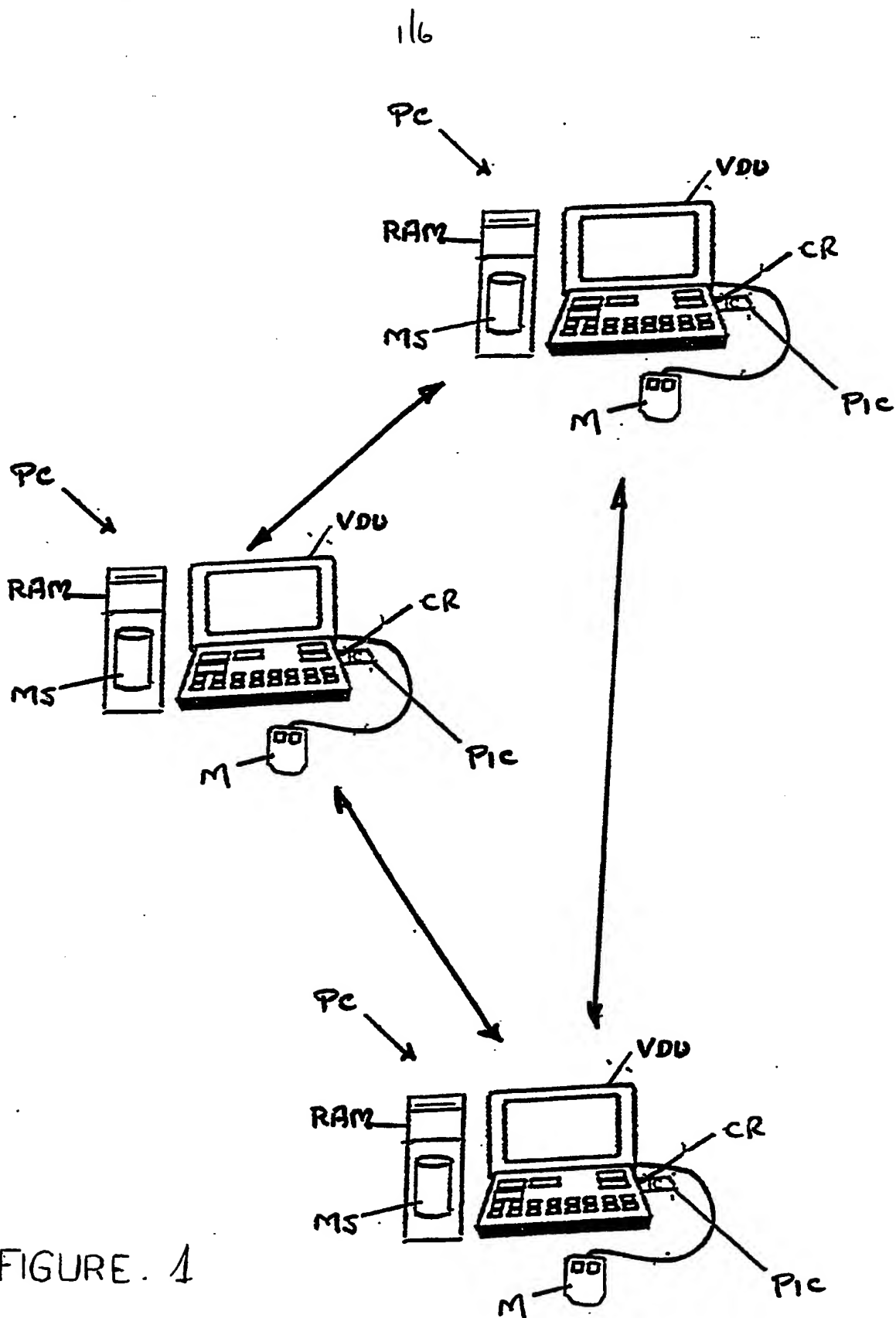


FIGURE. 1

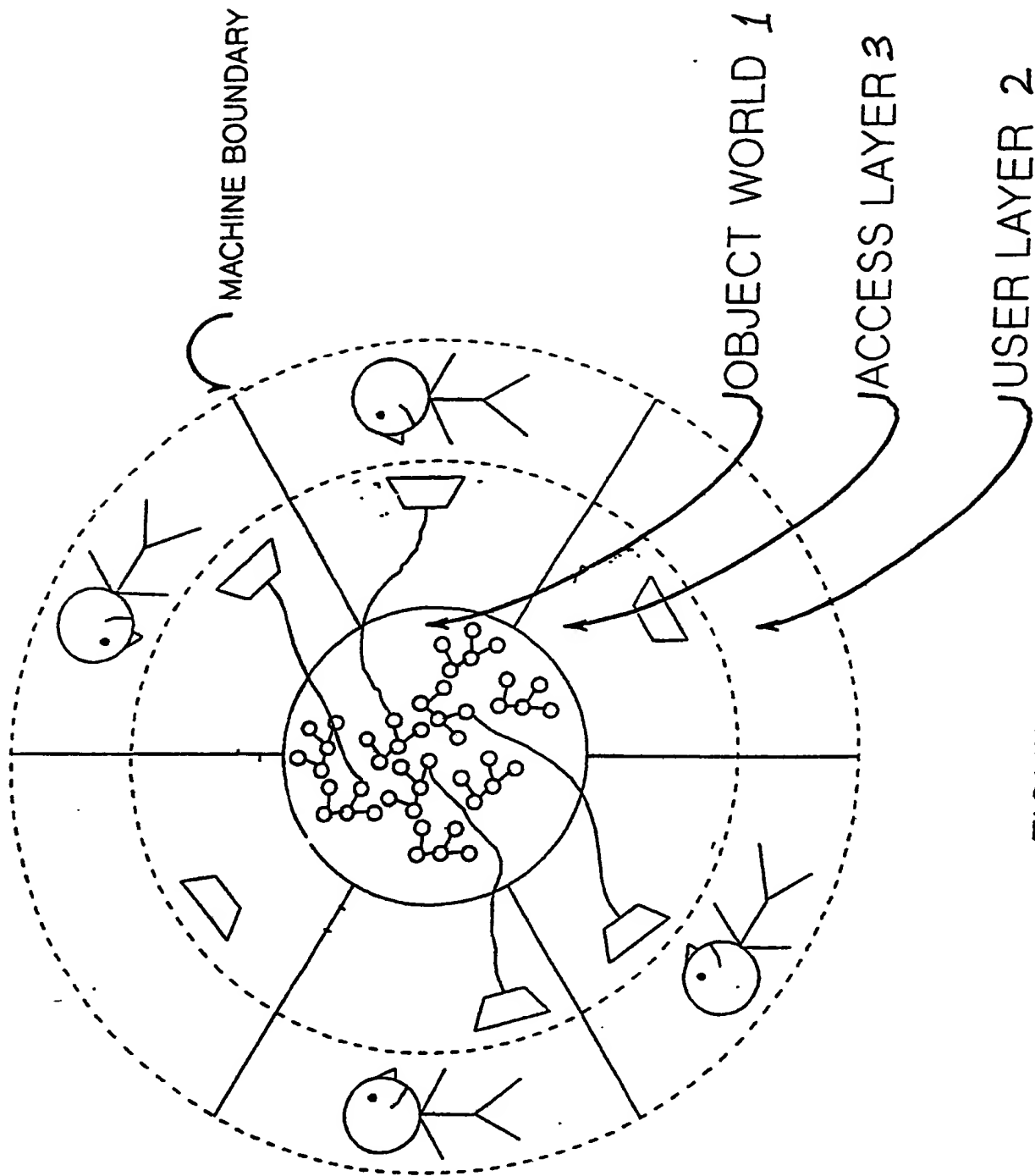


FIGURE 2

3/6

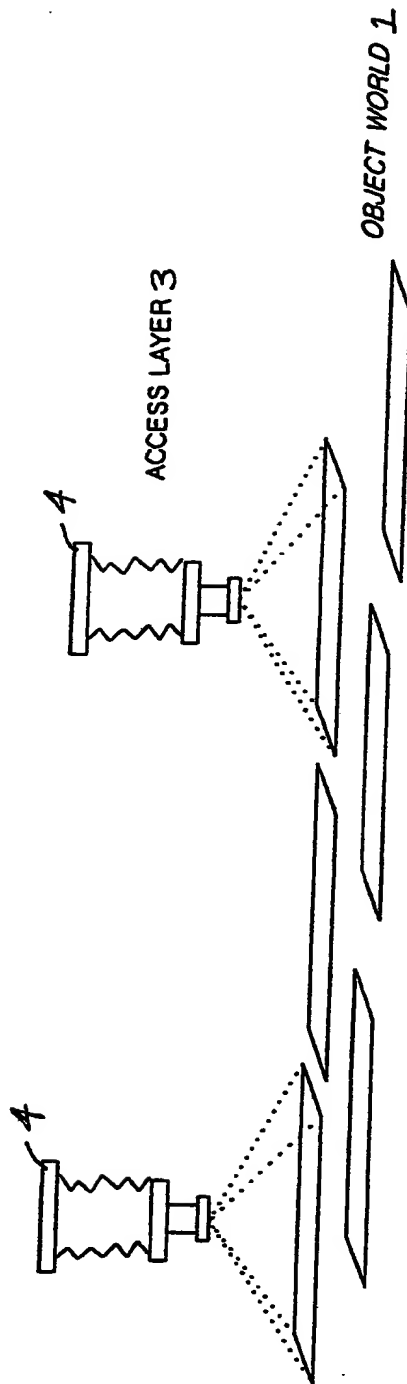


FIGURE 3



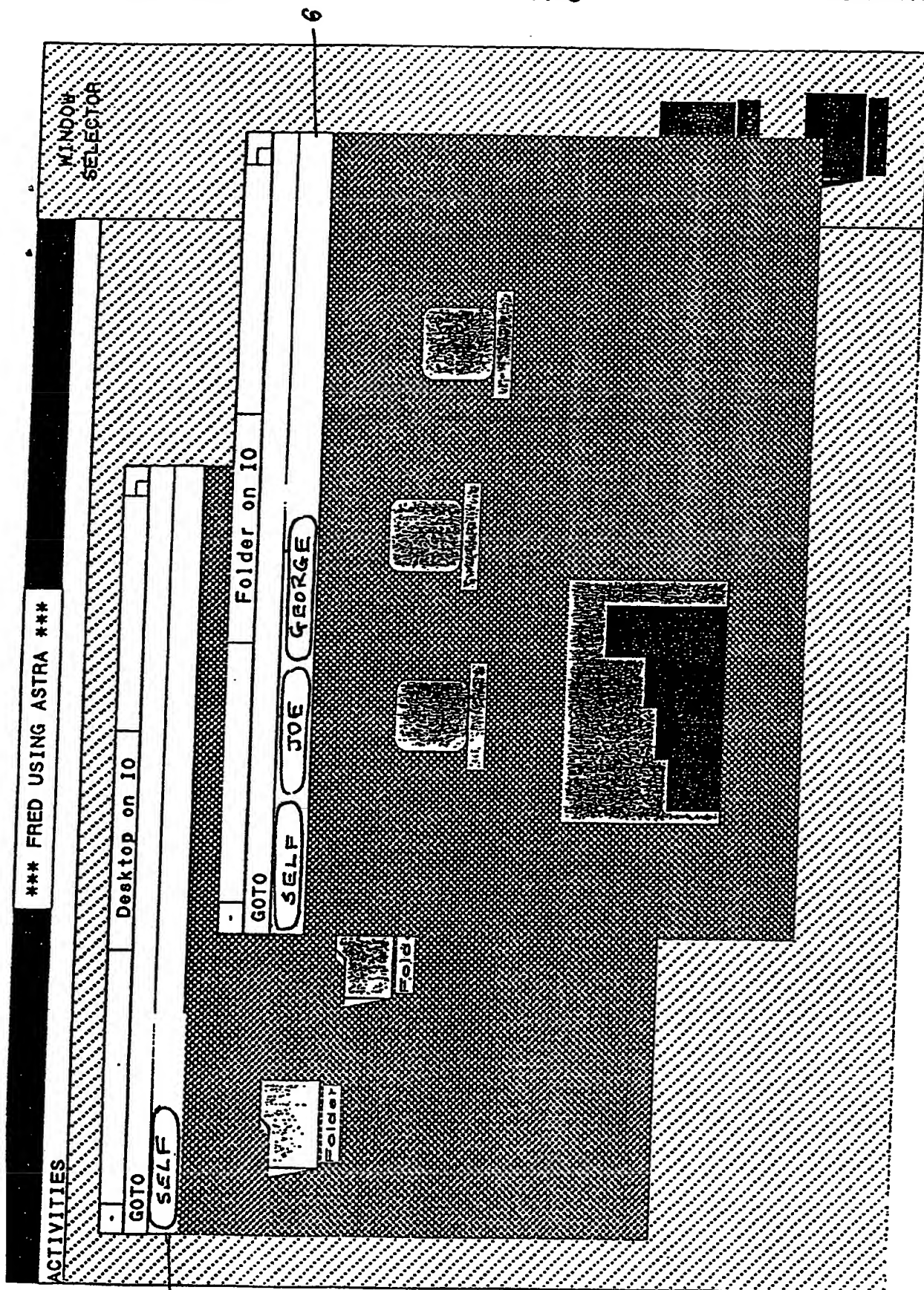


FIGURE 4

# SUBSTITUTE SHEET

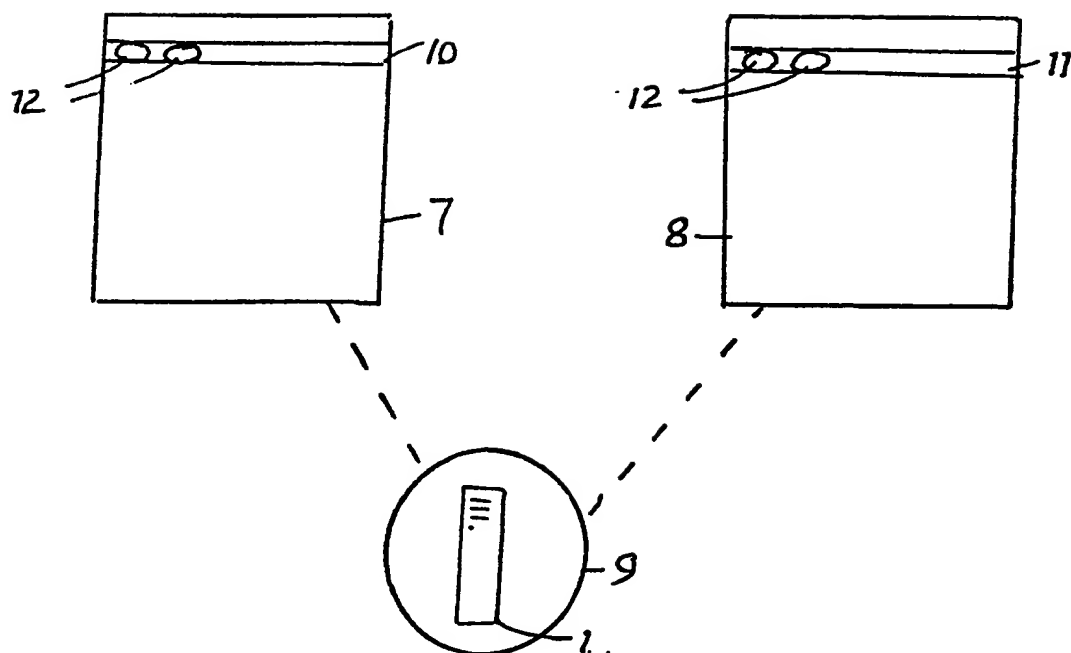


FIG. 5

616

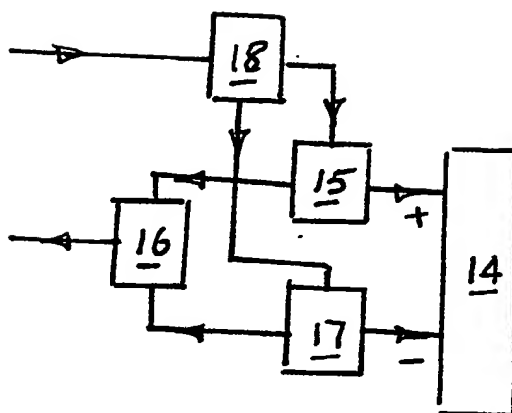


FIG. 6

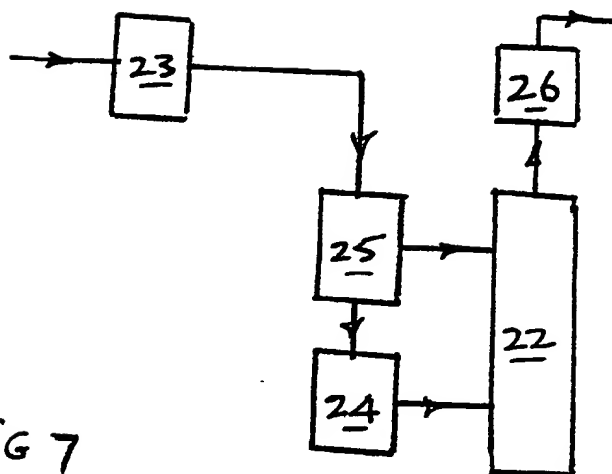


FIG 7

## INTERNATIONAL SEARCH REPORT

International Application No

PCT/GB 91/00687

<b>I. CLASSIFICATION OF SUBJECT MATTER</b> (If several classification symbols apply, indicate all) <sup>6</sup>		
According to International Patent Classification (IPC) or to both National Classification and IPC		
Int.Cl. 5      G06F9/44		
<b>II. FIELDS SEARCHED</b>		
Minimum Documentation Searched <sup>7</sup>		
Classification System	Classification Symbols	
Int.Cl. 5	G06F	
Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched <sup>8</sup>		
<b>III. DOCUMENTS CONSIDERED TO BE RELEVANT<sup>9</sup></b>		
Category <sup>10</sup>	Citation of Document, <sup>11</sup> with indication, where appropriate, of the relevant passages <sup>12</sup>	Relevant to Claim No. <sup>13</sup>
X	PROC. FROM THE 2ND INTL. CONF. ON EXPERT DATABASE SYSTEMS vol. XIII, April 25, 1988, TYSONS CORNER, VA, USA pages 107 - 131; A. MOTRO ET AL: 'The Design of KIVIEW: An Object-Oriented Browser' see page 110, line 2 - line 6 see page 119 see page 127, line 12 - page 129, line 11	1
Y	----	2
A	EP,A,328 827 (IBM CORP.) August 23, 1989 see page 2, line 32 - line 36 see page 6, line 35 - line 46 see page 8, line 8 - line 18	1-4
Y	----	2
A	----	1-4
<p><sup>10</sup> Special categories of cited documents:</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>"&amp;" document member of the same patent family</p>		
<b>IV. CERTIFICATION</b>		
Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report	
29 AUGUST 1991	27.09.91	
International Searching Authority	Signature of Authorized Officer	
EUROPEAN PATENT OFFICE	Brandt, J. E. <i>Brandt</i>	

III. DOCUMENTS CONSIDERED TO BE RELEVANT (CONTINUED FROM THE SECOND SHEET)		
Category *	Citation of Document, with indication, where appropriate, of the relevant passages	Relevant to Claim No.
A	<p>PROC. FROM THE 24TH ACM/IEEE DESIGN AUTOMATION CONFERENCE. 1987, MIAMI BEACH US pages 335 - 340; S.HEILER ET AL: 'An Object-Oriented Approach To Data Management: Why Design Databases Need It. ' see page 336, left column, line 10 - line 20 see page 337, right column, line 1 - line 30 see page 339, right column, line 1 - line 25</p>	1-4

**ANNEX TO THE INTERNATIONAL SEARCH REPORT  
ON INTERNATIONAL PATENT APPLICATION NO.**

GB 9100687  
SA 47171

This annex lists the patent family members relating to the patent documents cited in the above-mentioned international search report.  
The members are as contained in the European Patent Office EDP file on  
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

29/08/91

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP-A-328827	23-08-89	US-A- 4965719 JP-A- 1211140	23-10-90 24-08-89